لنكتب سكربتاً بسيطاً (فاتح شهية) :

```
<"html dir = "rtl>
التحية لدي أهل الإسلام هي
?>
 Echo ("السلام عليكم ورحمة الله وبركاته")
<?
<html/>
```

قم بحفظ الملف باسم **echo.php**
ستعرض علينا عباره مكتوب فيها

التحية لدي أهل الإسلام هي السلام عليكم ورحمة الله وبركاته

شي بسيط أليس كذلك ؟

يتكون كود الــ**php** من نصوص و كود و علامات ولغة **html** وقد لاتحتوي على نصوص **html** .
لكي يعمل الكود يجب أن يكون إمتداد الملف **php** أو بأي إمتداد من إمتدادات الــ**php**
مثلاً **php3** و **phtml**
--
عندما تطلب صفحة في الإنترنت فإنك تجري اتصالاً مباشراً مع السيرفر هذه العملية تدعي **request** للسيرفر (يعنـــي طلبيـــة للســـيرفر) يقـــوم السيرفر بتفسير طلبك والبحث عن الصفحة المطلوبة ويرسل اليك الصفحة المطلوبة كجزء ممـــا يســـمي **response** (اســـتجابة) لمســـتعرض

الانترنت لديك يقوم بعدها المتصفح لديك بأخذ الكود الذى ارجع إليه ويقوم بتجميعه (compile) لكي يصبح صفحة صالحة للعرض هذه العملية التي حصلت تشبه نظرية العميل للخادم(client to server) بحيث أن المتصفح هو العميل والخادم هو السيرفر .
الخادم يقوم بعملية تخزين وترجمة وتوزيع البيانات بينما يقوم العميل (مستعرض الانترنت لديك) بالعبور الى السيرفر واحضار البيانات

بروتوكولات الانترنت :

لانريد هنا أن نذهب إلى التكلم عن تاريخ انترنت العتيق ، النقطة المهمة هي الشبكة المربوطة بنقاط nodes الانترنت صممت لكي تقوم بالحفاظ على المعلومات لكي يتم نقلها من مكان إلى آخر وهي تستخدم مجموعة من البروتوكولات مثل Tcp/Ip لكي يتم نقل البيانات عبر الشبكة .

بروتوكول Tcp/Ip

من مميزات هذا البروتوكول أنه بإستطاعته إعاده تمهيد طريقه للبيانات إذا تم خلل في نقطة أو مكان أثناء نقلها ويتم ذلك بسرعة شديدة.عندما يطلب المستخدم من المستعرض أن يجلب له صفحة من الانترنت فإن المستعرض يجلب هذه الأوامر باستخدام بروتوكول يدعي بروتوكول التحكم في نقل البيانات TCP هذا البروتوكول هو بروتوكول نقل للبيانات وهو يضمن أن البيانات قد تم إرسالها ووصولها بشكل صحيح .

قبل أن يتم إرسال البيانات عبر الشبكة يجب عنونتها والبروتوكول الذي يقوم بعنونة البيانات يدعي HTTP يقوم هذا البروتوكول بوضع عنونة للبيانات لكي يعرف البروتوكول TCP أين سينقل البيانات (فهو لايستطيع نقل البيانات إذا لم يكن لها هدف أو مكان) يستخدم البروتوكول HTTP عن طريق الويب في عملية نقل البيانات من كمبيوتر إلى آخرعندما ترى الصفحة متبوعة بـhttp:// فانك تعلم مباشرة أن الانترنت يستخدم البرتوكول HTTP لإحضار هذه الصفحة يمكنك أن تاخذ صورة بأن الـTCP عبارة عن ساعي بريد الذي يقوم بإيصال رسالة ، هذه الرسالة فيها طابع بريد وعنوان وهو مانسميه بالـHTTP .

يتم تمرير الطلب من المستعرض إلى ملقم أو سيرفر الويب وهو مايعرف بـ HTTP request ويقوم السيرفر برؤية مستودع البيانات لديه لكي يحصل على البيانات المطلوبة فإذا وجد الصفحة في المستودع قام بإرسالها على شكل حزم الى الجهة التي قامت بالطلب باستخدام بروتوكول TCP ويعنون هذه الحزم لمستعرض الانترنت لديك باستخدام بروتوكول http (ننبه دائما الى أنه يرسلها على شكل حزم لكي تعرف السبب عند

عدم ظهور صفحة ويب كاملة أن هناك حزمة لم ترسل بشكل جيد) ولكن إذا لم يجد السيرفر الصفحة المطلوبة فانه يقوم بإرسال صـــفحة تحتـــوي على رسالة خطا **404** وهذه الصفحة التي أرسلت من ملقم الويب الى المستعرض لديك تسمي **HTTP response** .

بروتوكول الــ HTTP

رغم ما أخذناه من معلومات كثيرة وقصص كثيرة تشبه قصص ألف ليلة أو حكايات الأطفال إلا أنه رغم ذلك يفوتنا الكثير من التفاصيل فـــي هـــذا الموضوع لذلك دعنا نغوص قليلاً في التفاصيل عن بروتوكول **HTTP** بشكل خاص.

عندما تقوم بعملية طلب لصفحة من السيرفر هناك أمور إضافية ترسل مع عملية الطلب **http request** غير الـــ**URL** وهي ترسل كجزء من **http request** .

نفس الموضوع مع ال**http response** هناك أمور أخرى تصل معه كجزء منه .

الكثير من هذه المعلومات تولد تلقائياً في رسالة الــ**HTTP** ولايقوم المستخدم بالتعامل معها مباشرة , إذن لايحتاج أن تقلق نفســك بشــأن هــذه المعلومات إذا أنت لم تنشأها في الأصل ويجب أن تأخذ أيضا في معلوماتك أن هـــذه المعلومـــات ترســل كجــزء مــن الــــ**HTTP request** والـــ**HTTP response** لأن سكربت الــ**PHP** الذي نصنعه يمنحنا تحكماً إضافيا بهذه المعلومات .

كل رسائل الــ**HTTP** تأخذ تنسيقاً معيناً سواء كانت **Request** أو **Response** . نستطيع أن نقوم بتقسيم هذا التنسيق إلى ثلاثة أقسام :

1 – **Request/response line**

2 – **Http header**

3 – **Http body**

المحتوي من هذه الأشياء الثلاثة يعتمد على نوع الرسالة إذا كانت **HTTp Request** أو **HTTP response** لذلك سنتكلم عنهم بتعمق أكثر .

Http Request

يجب أن يحتوي الــ**request** على الأقل الــ**request line** (سطر الطلب) والــ**HOST** .
يرسل مستعرض الانترنت طلبية (**HTTP request**) إلى ملقم الويب تحتوي على التالي :

1- The Request Line

السطر الأول من كل طلبية (**http request**) هي **Request Line** الذي يحتوي على ثلاثة أنواع من المعلومات :
أ – أمر **HTTP** وهو مايعني بــ **method** .
ب – المسار من السيرفر إلى المصادر المطلوبة (صفحات الانترنت) المطلوبة من قبل العميل (المستعرض)
جـ – إصدارة الــ**HTTP** .

إذن كمثال على الــ **Request Line** أنظر إلى السطر التالى :

GET /testpage.htm HTTP/1.1

الــ**method** يخبر السيرفر كيف يتعامل مع الطلب هناك ثلاثة أنواع شائعه من الــ**method**

2- HTTP Header

البت الثاني من المعلومات هو الهيدر **HTTP Header**. الذي يحتوي على تفاصيل أو وثائق عن العميل مثــل نــوع المتصــفح (نتســكيب أو إكسبلور) الذي قام بطلب الصفحة والوقت والتاريخ والإعدادات العامة
الــ**HTTP Header** يحتوي على معلومات نستطيع تقسيمها الى ثلاث فئات وهي :
أ – عامة **GENERAL** : تحتوي معلومات إما عن العميل أو السيرفر ولاتخصص إلى فرد أو مجموعة .

ب – شخصية **Entity** : تحتوي على معلومات عن البيانات التي أرسلت بين المتصفح والسيرفر .
جـ – مطلوبة **Request** : تحتوي على بيانات عن إعدادات العميل والأنواع المختلفة المقبولة من البيانات .

وهذا مثال :

Accept: * / *

Accept language: Arabic-KSA.
Connection: Keep –Alive.
Host : http://www.arabbuielder.com
Referer: http://www.arabbuielder.com/index.php?something=132
User –Agent :Iexploer (win98;.........)

مثلما ترى الـــ **HTTP Header** عبارة عن إعداد يتكون من عدة سطور كل سطر يحتوي على قيم معينة .

هناك عدة سطور تشكل الـــ **HTTP header** وأكثرها إختياري , يقوم الـــ**HTTP** بالإخبار عن إنتهاء معلومات الـــ **header** بتـــرك ســـطر فارغ (وهذا يكون في الـــ**HTTP1.1**) .

3– The HTTP Body :
إذا تم استخدام الأمر **POST** في الـــ**HTTP Request Line** عندها يقوم الـــ **HTTP** بطلب المعلومات التي ارسلت في الـــــ**body** الـــى السيرفر .

Http Response
يرسل من السيرفر إلى المستعرض ويحتوي على ثلاثة أشياء :
1– the Response Line

http header – 2
Http Body – 3

The Response Line – 1

الـ **response line** يحتوي فقط على نوعين من المعلومات :

1 – رقم إصدارة الـ**HTTP** .

2 – شفره أو كود الـ**http request** التي تقوم بتحديد إذا كان الـ**request** ناجحاً أم فاشل .

مثال :

HTTP/1.1 200 OK

في هذا المثال يقوم الـ**line response** بإرجاع القيمه **200** متبوعة بالكلمة **OK** هذه تشكل وتشير إلى نجاح الـ **request** ويكون الـ**response** يحتوي على الصفحة المطلوبة والبيانات من السيرفر . ومثال آخر هو الشفرة **404** عندما تقوم بطلب صفحة ويفشل السيرفر في الحصول عليها .

HTTP Header – 2

الـ **response header** يعتبر مشابه **request hader** الذي ناقشناه في الأعلى .وتنقسم المعلومات التي فيه أيضا إلى ثلاثة أنواع :

أ – عامة **GENERAL** : معلومات عن الـ **client** أو السيرفر ولاتخصص إلى واحد منهما .

ب – شخصية **Entity** : يحتوي على معلومات عن البيانات التي يتم ارسالها بين السيرفر والعميل .

جـ – الإجابة **Response** : يحتوي معلومات عن السيرفر الذي قام بإرسال الرد وكيفية تعامله ومعاجلته للرد (**Response**) .

كما قلنا سابقاً ، يتكون من عده سطور ويتم وضع سطر فارغ للإعلام عن إنتهاء الهيدر .

مثال :

HTTP/1.1 200 OK -the satus line
Date: Mon; 1st Nov 1999, 16:12:23 GMT -general header
Server : Apache/1.3.12 (Unix) (SUSE/Linux) PHP/4.0.2 -the response
Last-modified: Fri, 29 Oct 1999, 12:08:03 GMT -Entity Header

السطر الأول ناقشناه والسطر الثاني مفهوم من غير شرح ، السطر الثالث يقوم بتحديد البرنامج تبع السيرفر ونوعه ونظام التشغيل القــائم عليــه والسطر الأخير يقوم بتعريف آخر وقت تم فيه تعديل أو تجديد الصفحة .

ملاحظة : قد يحتوي الهيدر على أكثر من هذه المعلومات أو معلومات مختلفة وهذا يعتمد على نوع الشي المطلوب من السيرفر .

Http Body – 3

إذا تم معالجة الطلب بنجاح ، فإن الــ**HTTP response Body** يحتوي على كود الــ**HTML** ويقوم مستعرض الانترنت بتفسيرها وتحويلها إلى الصفحة النهائية التي تراها .

أين سكربت الــPHP من ذلك كله ؟

أصبح الآن لدينا مفهومية جيدة عن طريقة إرسال المستعرض طلب صفحة من السيرفر وكيفية استجابة السيرفر لهذا الطلب .

تكلمنا عن أن سكربت الــ**php** يتكون من ثلاثة أشياء : نص وكود **php** وكود **html** ، لانسطيع وصف الــ**html** بأنها لغة برمجة بشكل جيد ونستطيع أن نقول أن الــ**php** لغة سكربتات **Scripting Language** لأنها تضيف قدرات **html** عليها مثل الجــداول والفريمــات بكــود **html** بداخل كود الــ**php** هناك لغات تسمي لغات سكربتات قد تكون متآلفاً معها مثل الجافا سكربت والفجول بيسك سكربت بإستثناء أن الفــرق بينها وبين الــ**php** هو أن الــ**php** لغة تعتمد على جهة المزود أي السيرفر ويمكنك تخصيص المتصفح الذي يستعرضها .
تجعلنا الــ**html** نضمن سكربتات الــ**php** فيها ضمن قواعد لذلك لكي نستطيع تشغيلها ولكننا لاننسي أن إمتداد الملفات يظل كما هــو **php** أو **php3** بدون تغير فيه لكي يتم إرسال السكربت الى مكتبة الترجمة (**scripting engine**) التي تقوم بترجمة السكربت إلــى **html** (كأنــك تترجم من عربي لإنجليزي أو العكس)

مفهوم الــparsing و الــExecution :

ممكن أن نقسم عملية الترجمة الذي يقوم بها سيرفر php إلى قسمين أو عمليتين :

العملية الأولى : هي أن السيرفر يقوم أولا بفحص قواعد اللغة وهذا لايضمن أن السكربت صحيح مائة بالمائة ولكنه تدقيق في الأوامــر وقواعــد اللغة وهذا مايسمونه بالــParsing

العملية الثانية : هي تنفيذ السكربت بعدها وإخراجه على شكل كود html وهذا مايسمي بالــ Execution .

بقي أن نقول أمراً معروفاً وهو أن السكربتات نوعين :

1 – وهو ماينفذ من جهة المزود
Server –Side scripting
2 – ماينفذ من جهة المستعرض (صفحة انترنت) .

Mimoune zakaria